

FIG. 1

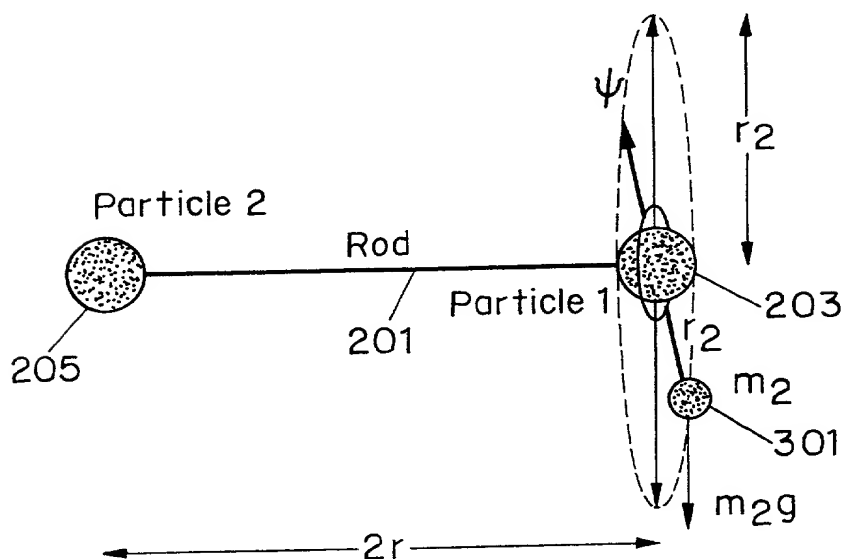


FIG. 3

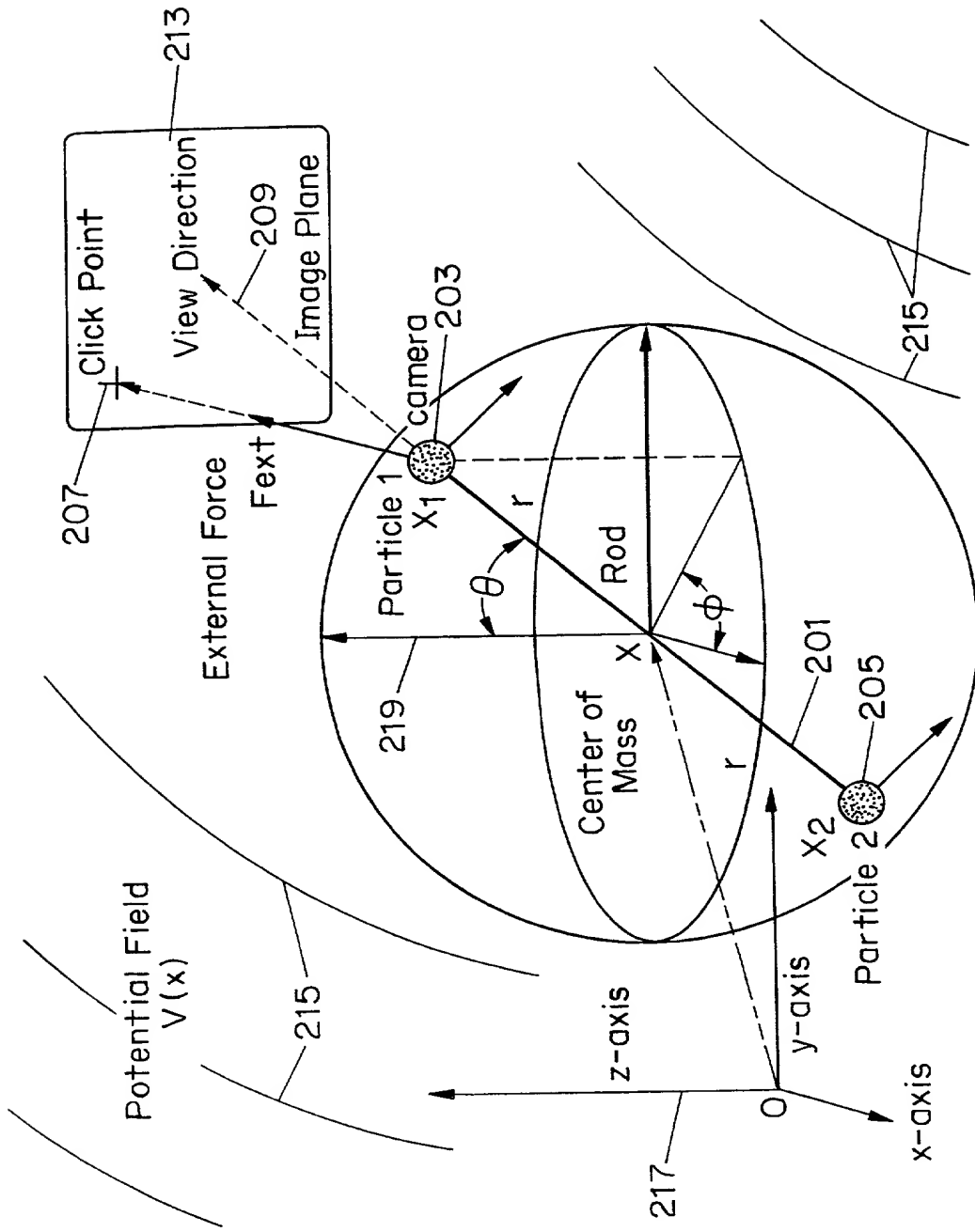


FIG. 2

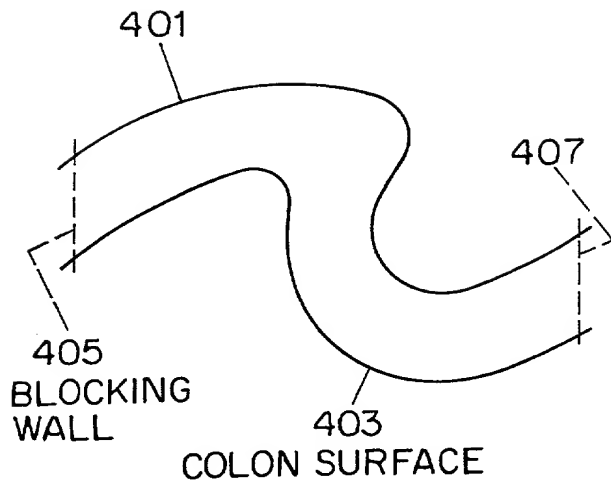


FIG. 4

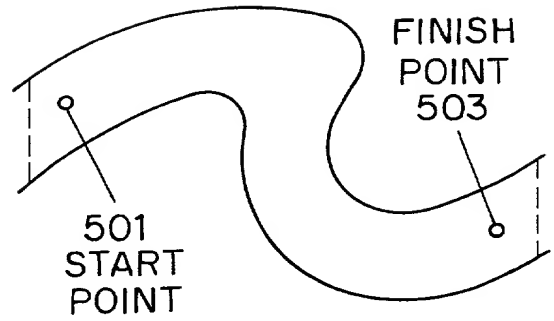


FIG. 5

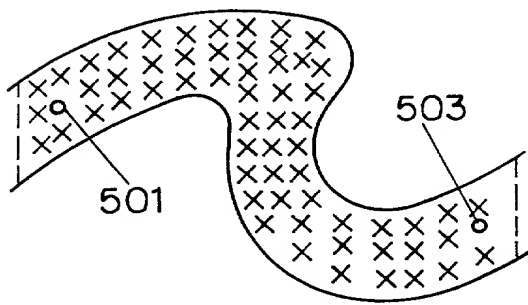


FIG. 6

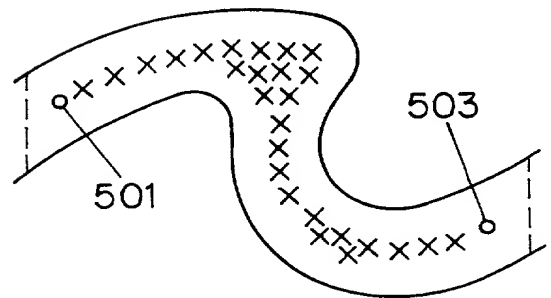


FIG. 7

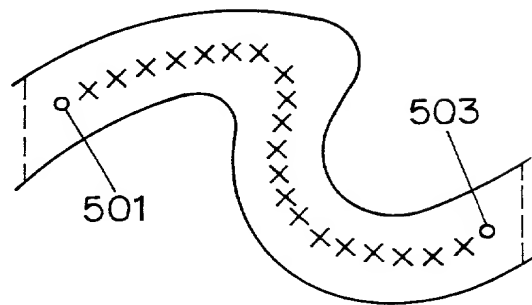


FIG. 8

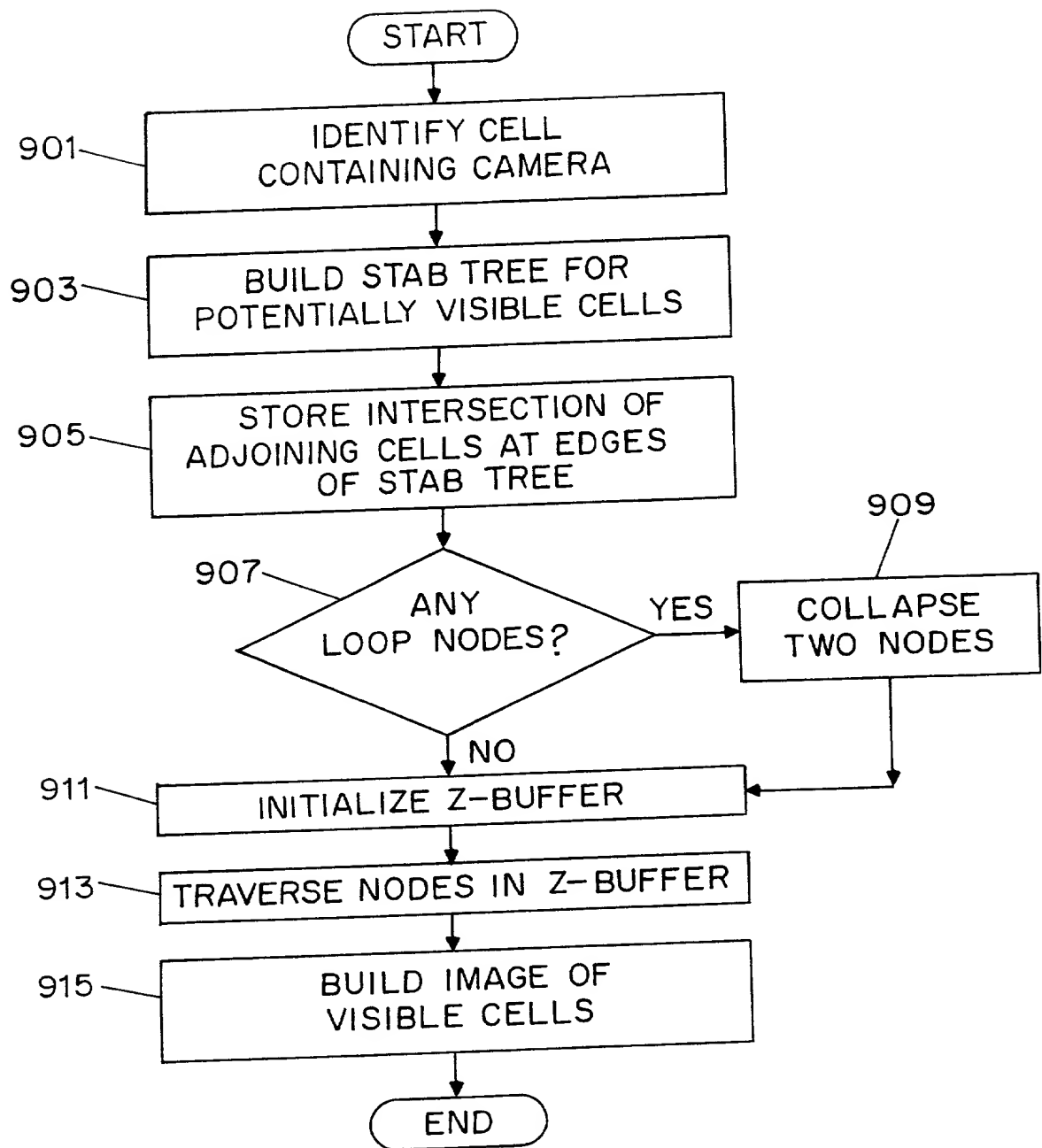


FIG. 9

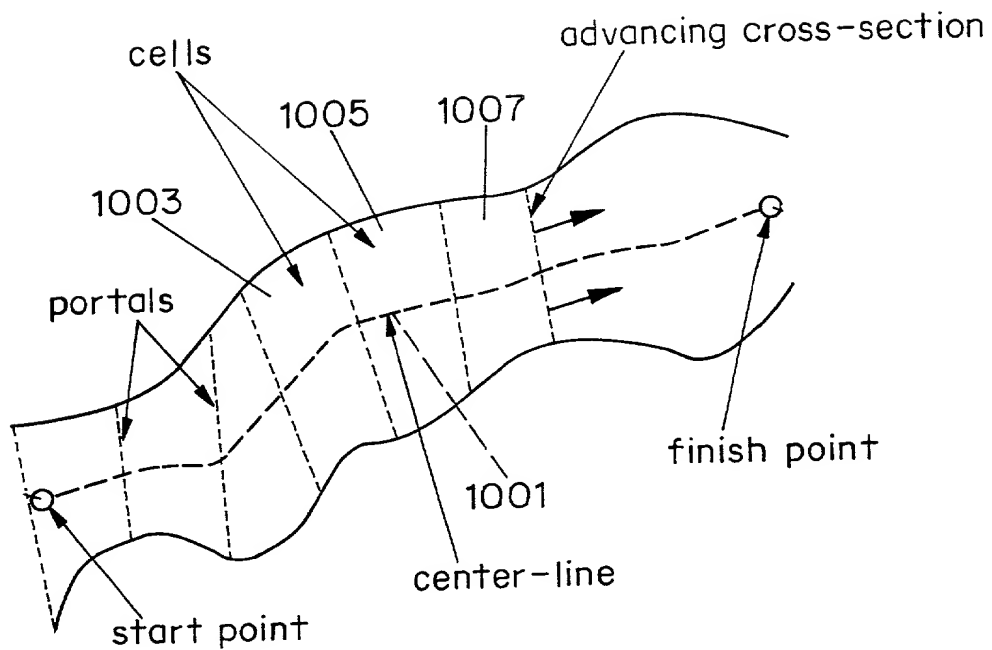


FIG. 10

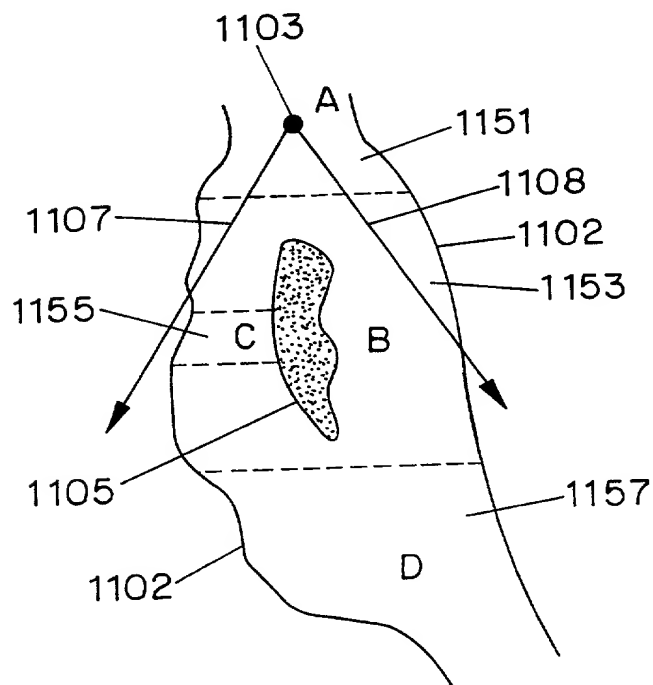


FIG. 11(a)

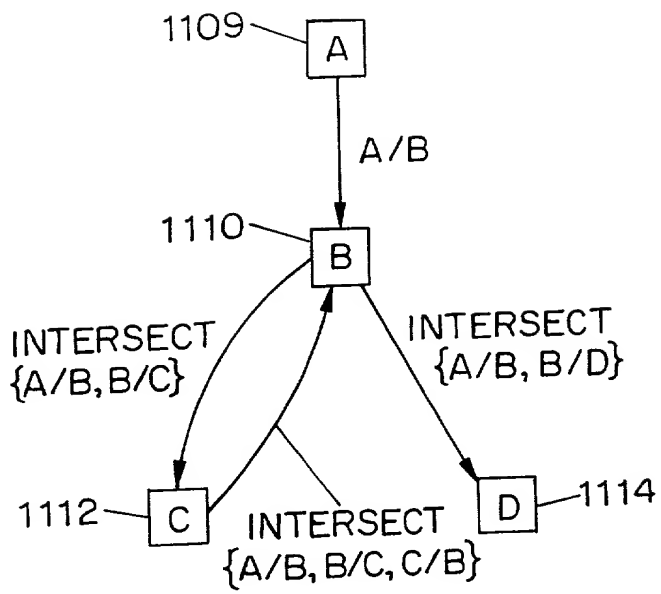


FIG. 11(b)

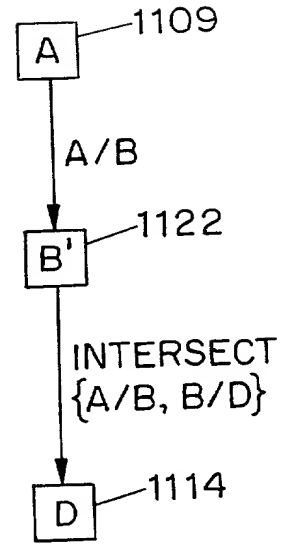


FIG. 11(c)

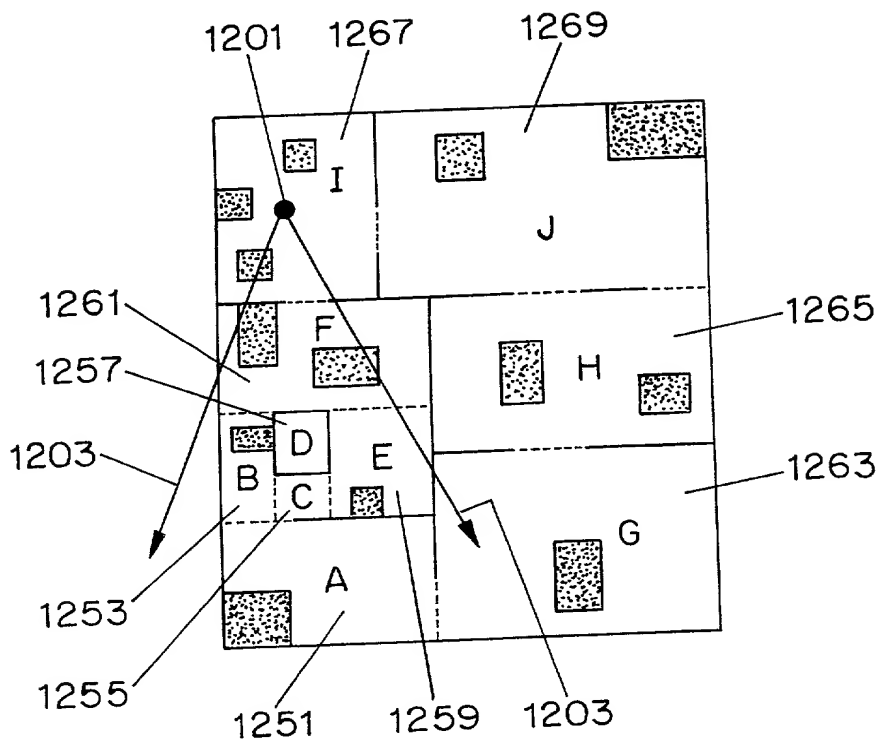


FIG. 12(a)

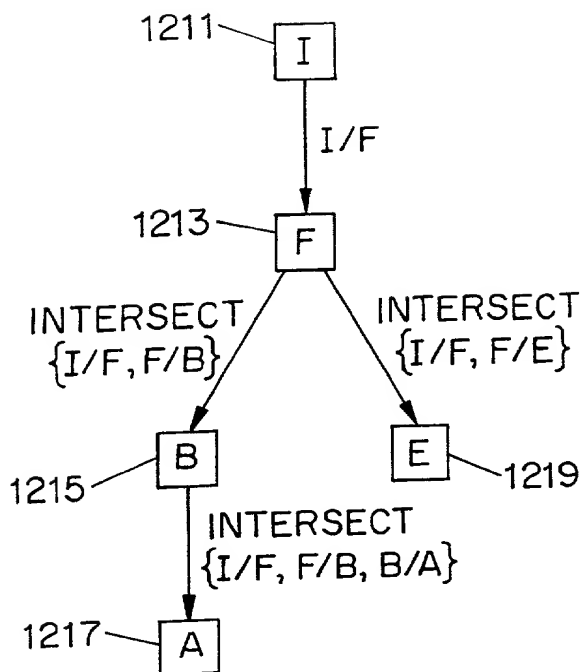
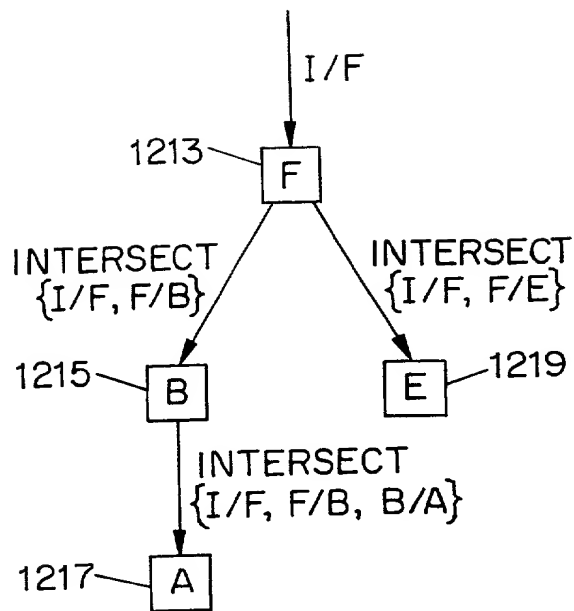


FIG. 12(b)



RENDERED NODES {I}
SKIPPED NODE { }

FIG. 12(c)

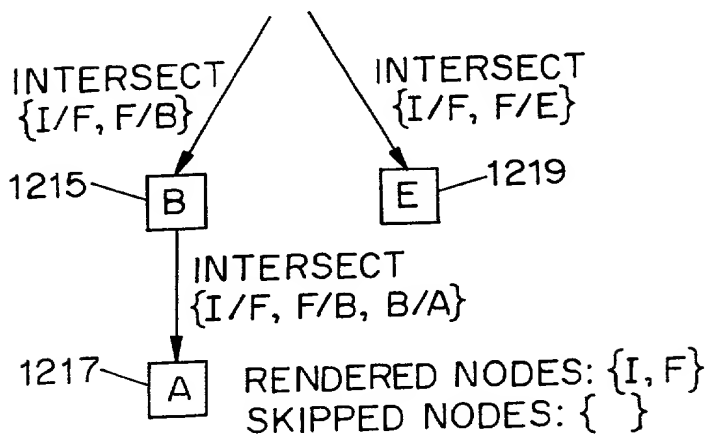


FIG. 12(d)

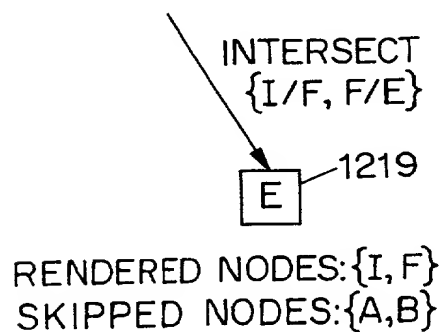


FIG. 12(e)

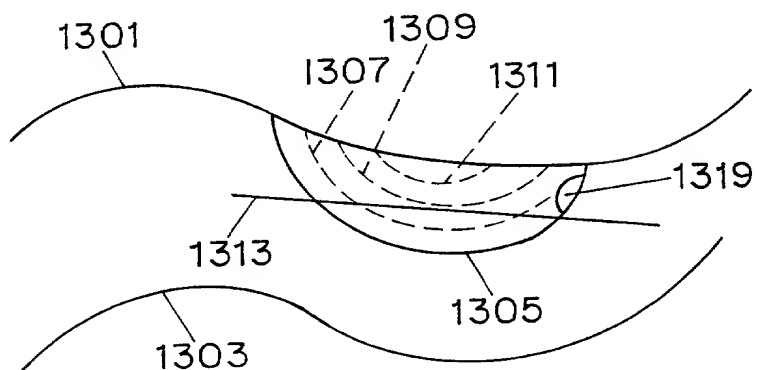


FIG. 13

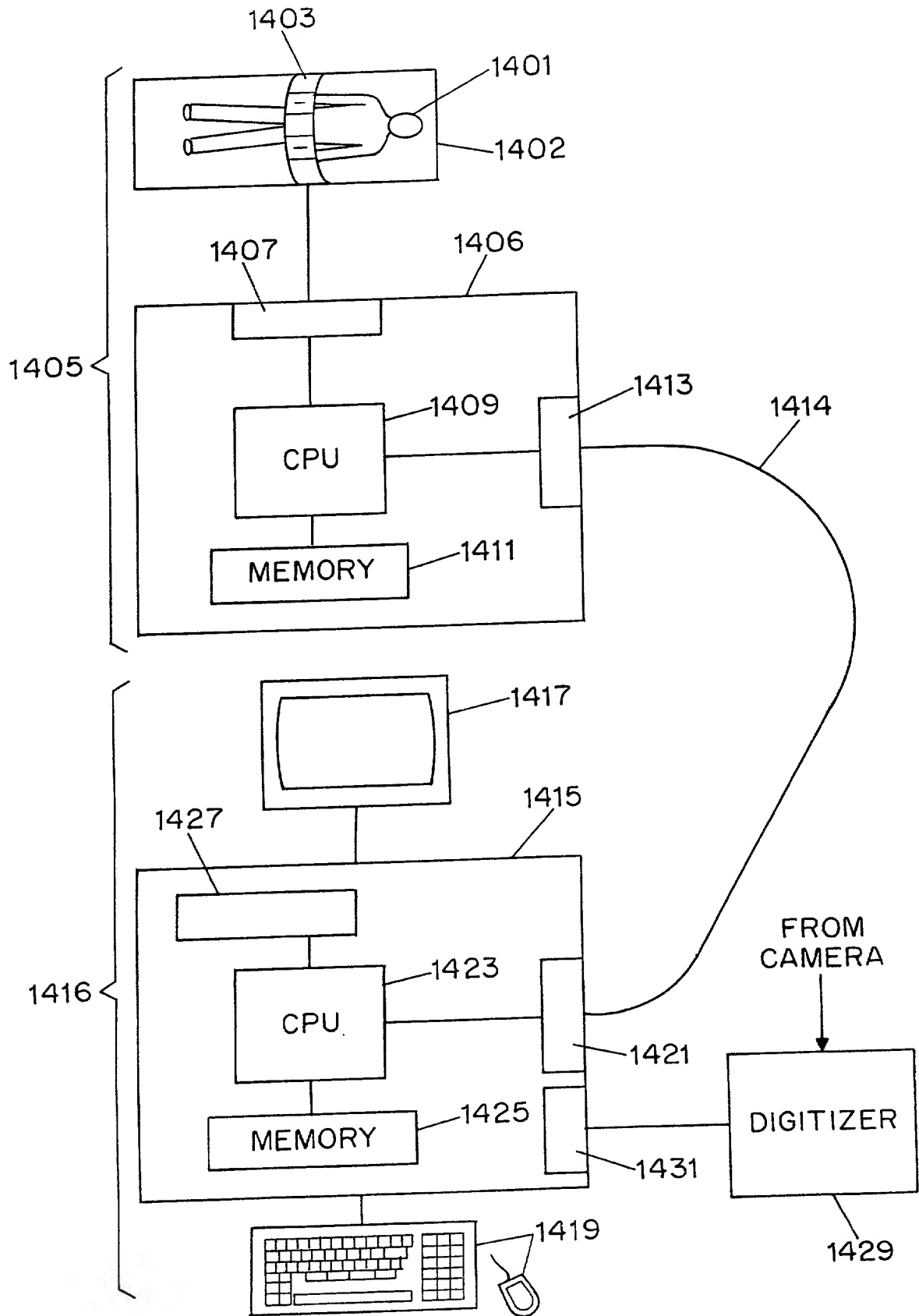


FIG. 14

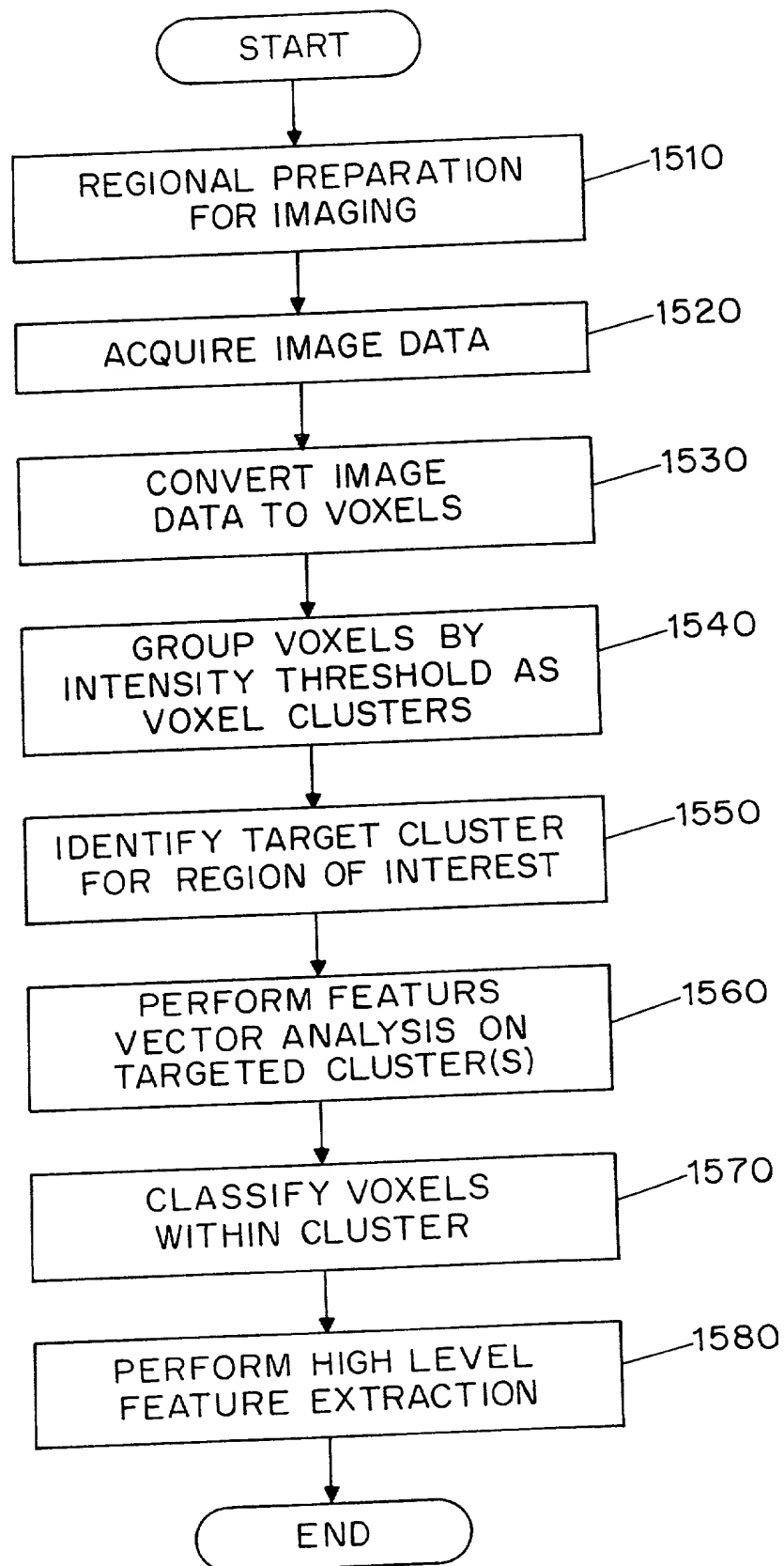


FIG. 15

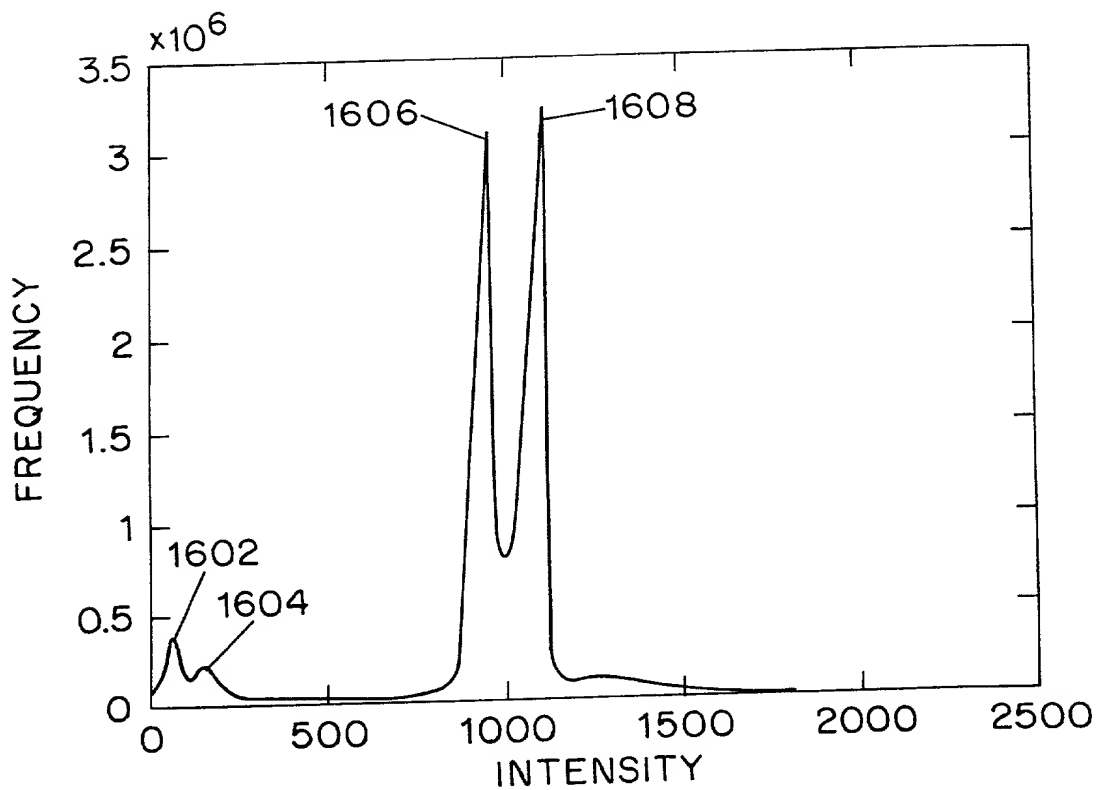


FIG. 16

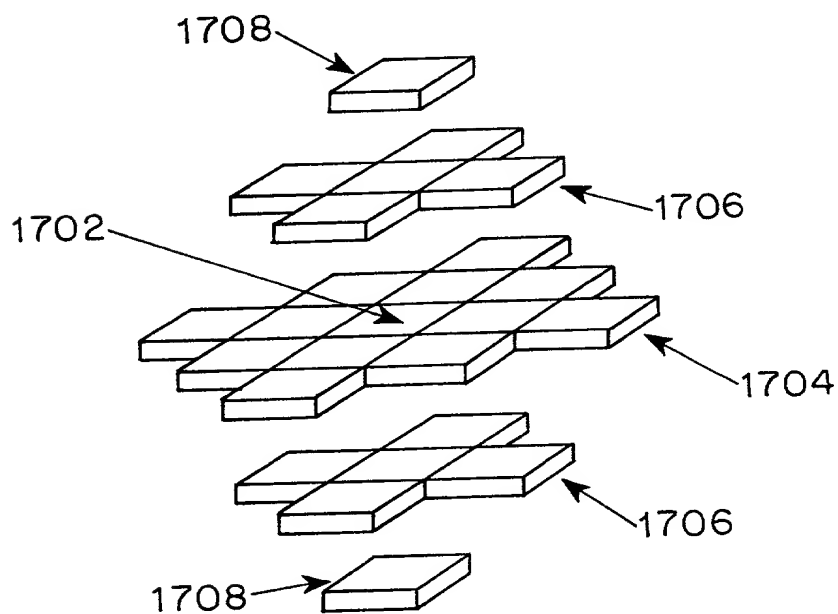


FIG. 17



FIG. 18A

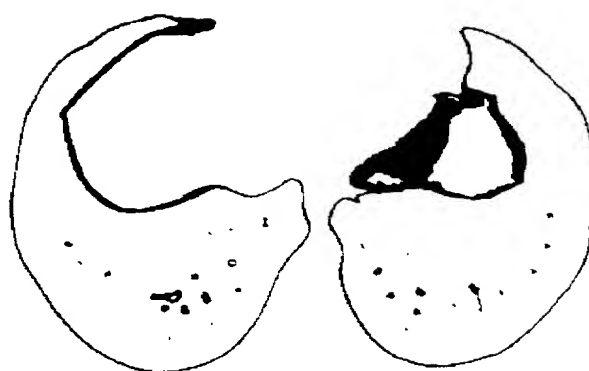


FIG. 18B

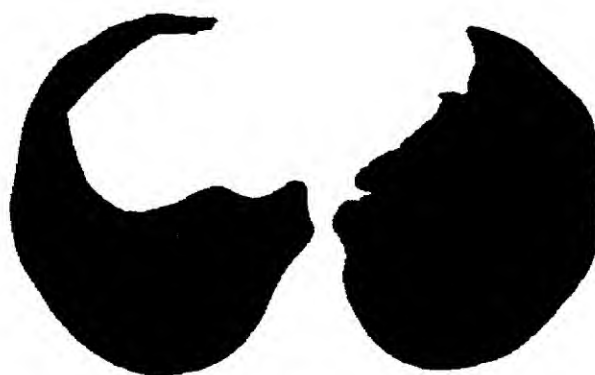


FIG. 18C

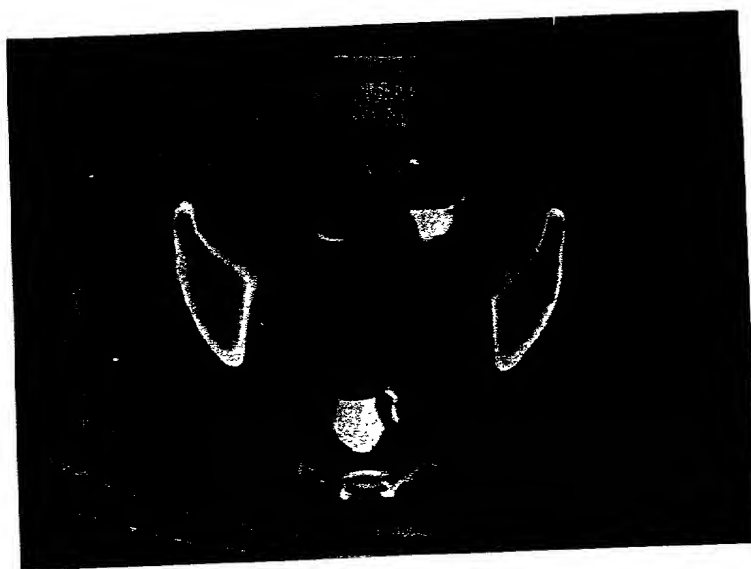


FIG. 19A

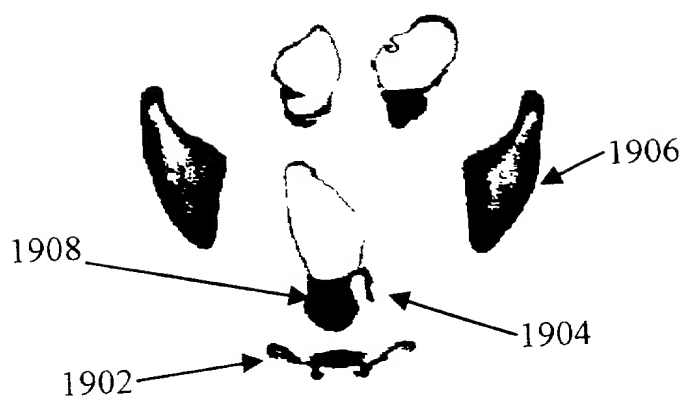


FIG. 19B



FIG. 19C

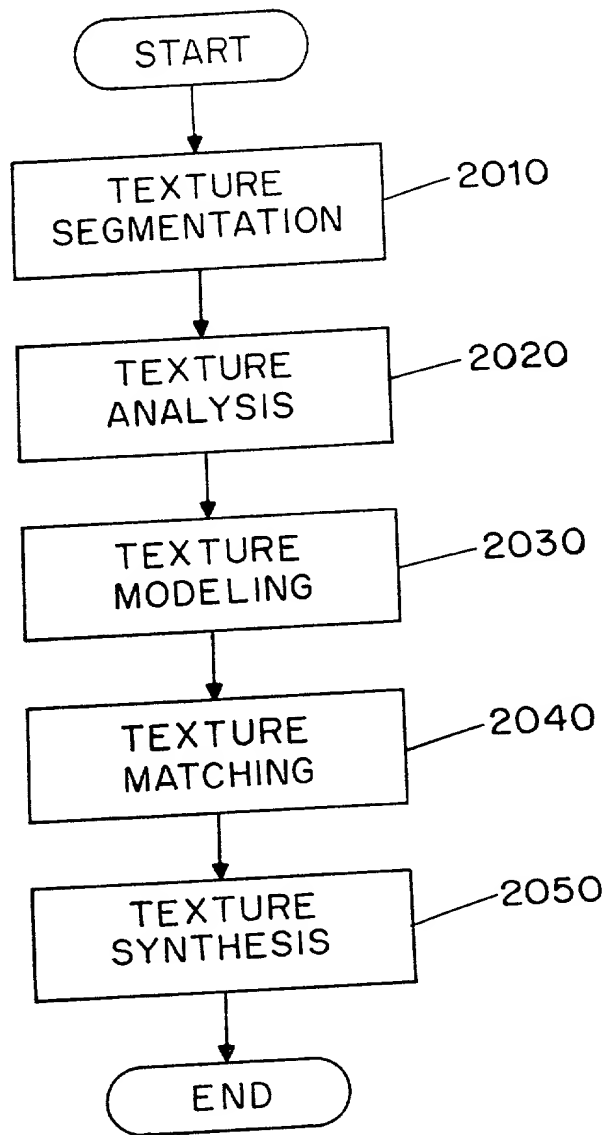


FIG. 20

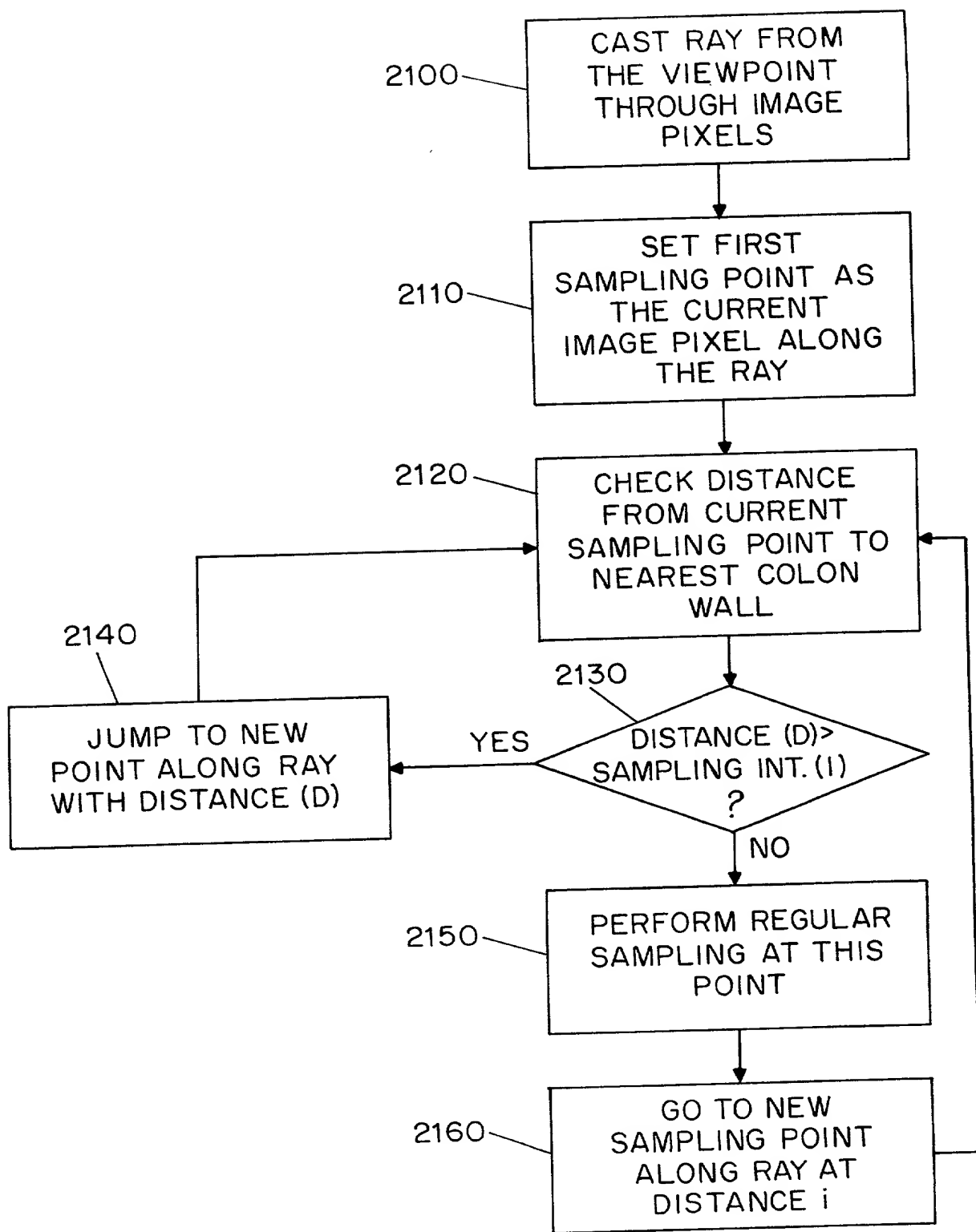


FIG. 21

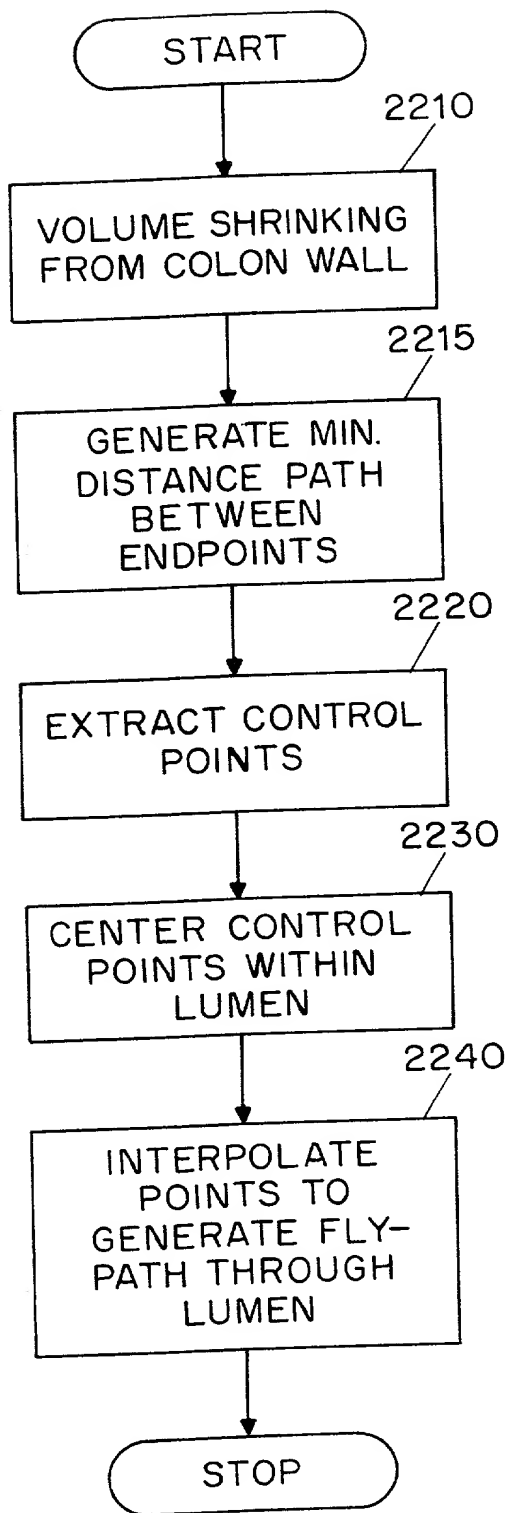


FIG. 22

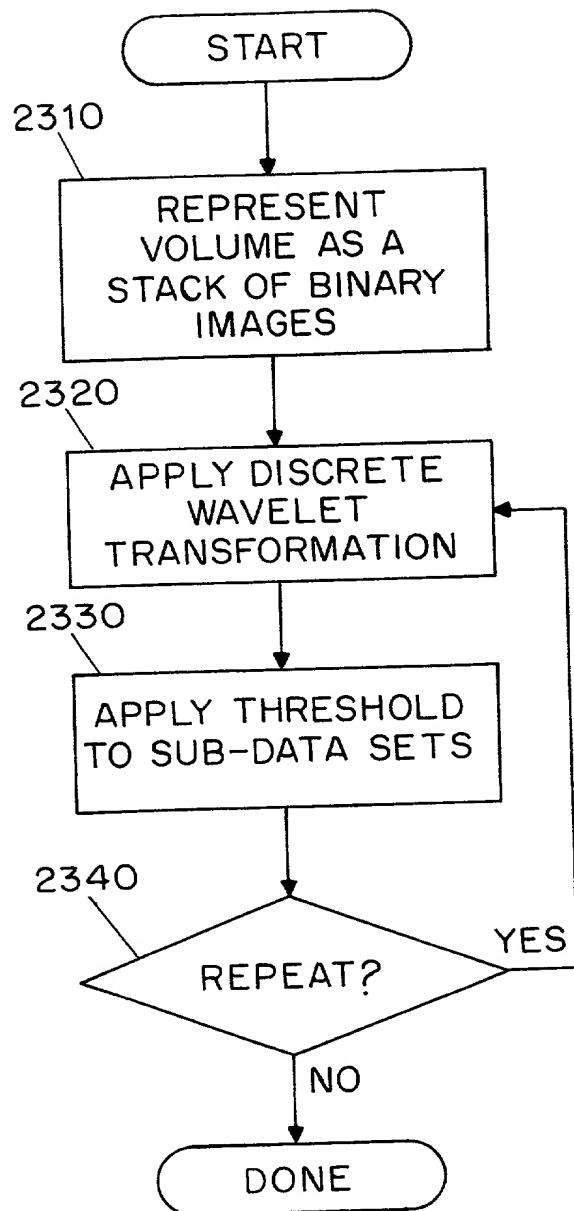


FIG. 23

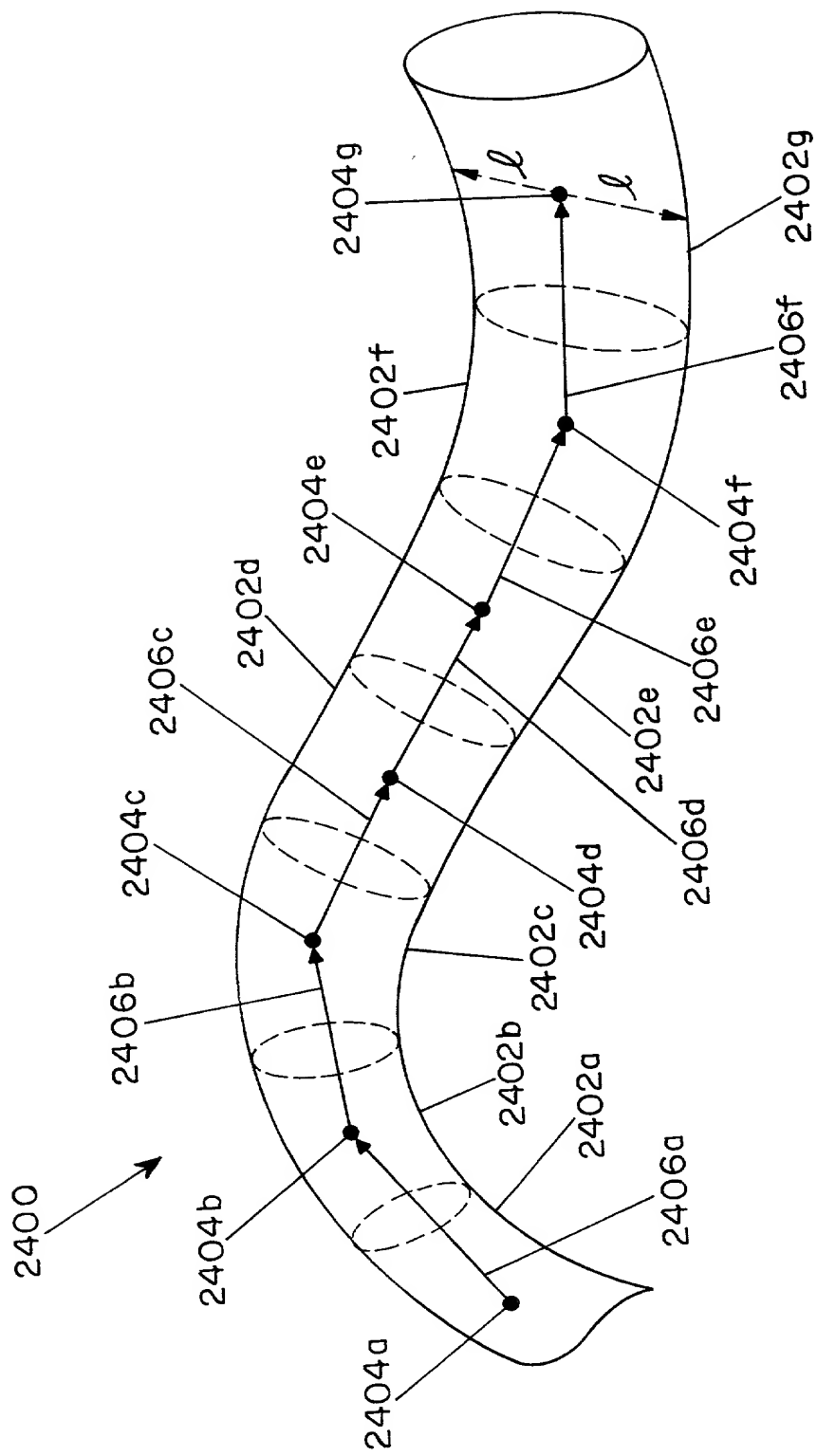


FIG. 24

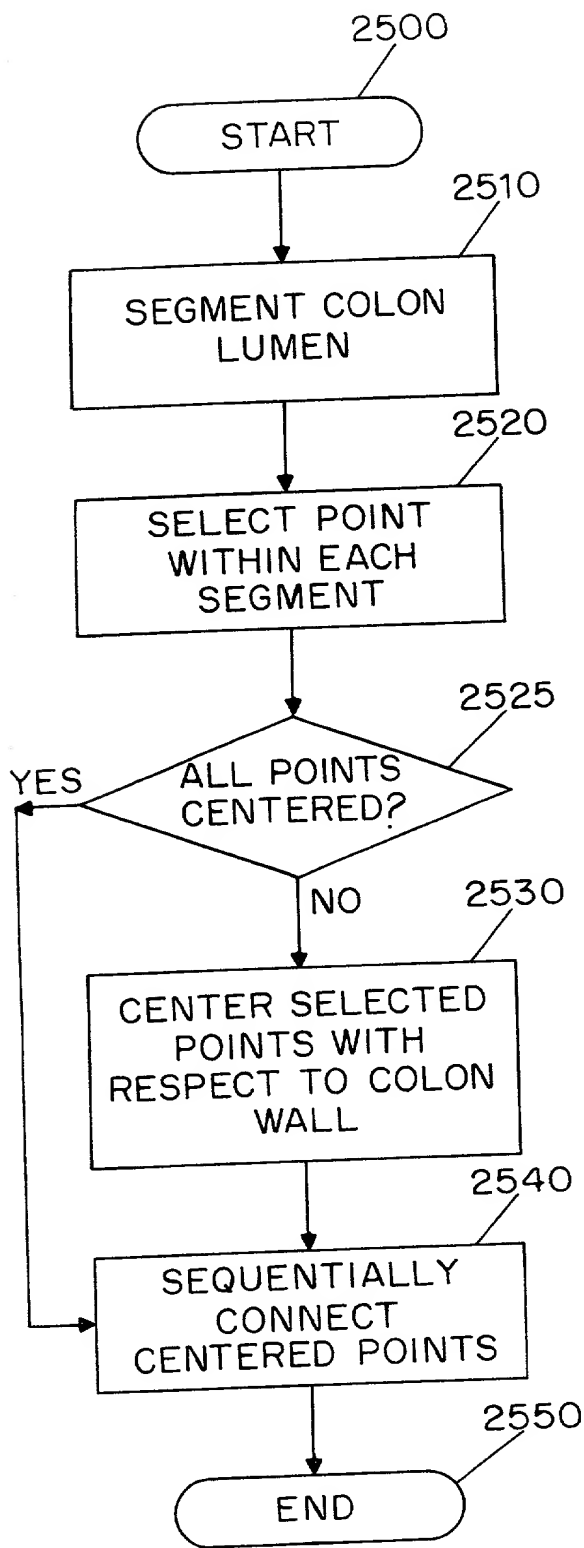


FIG. 25

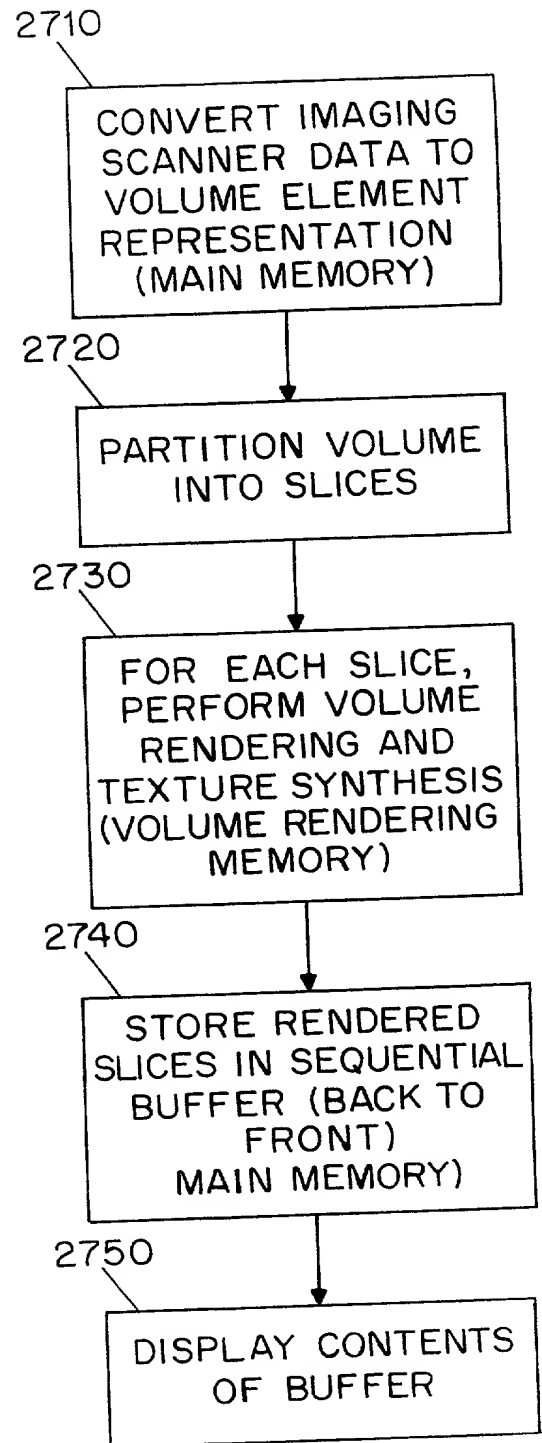


FIG. 27

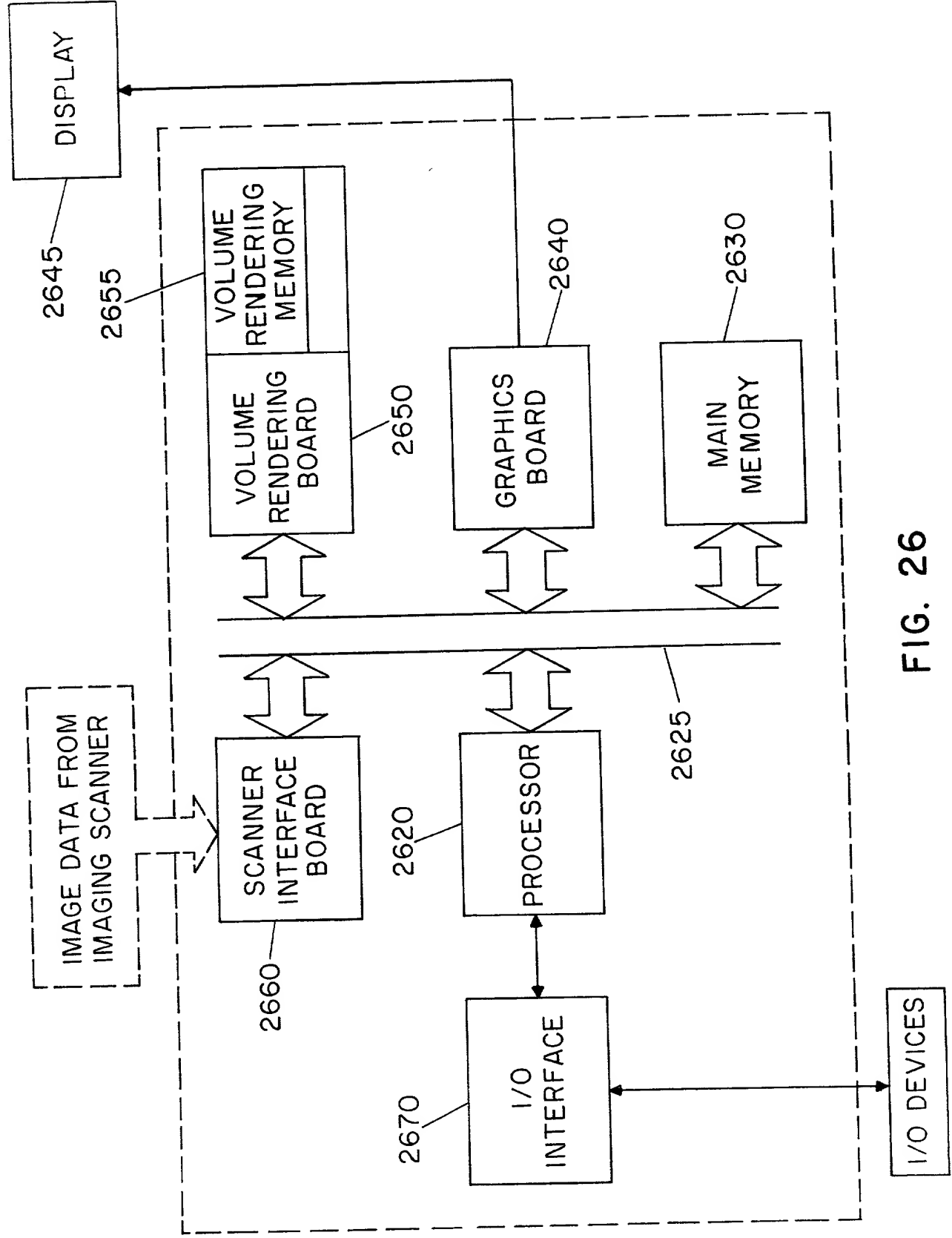
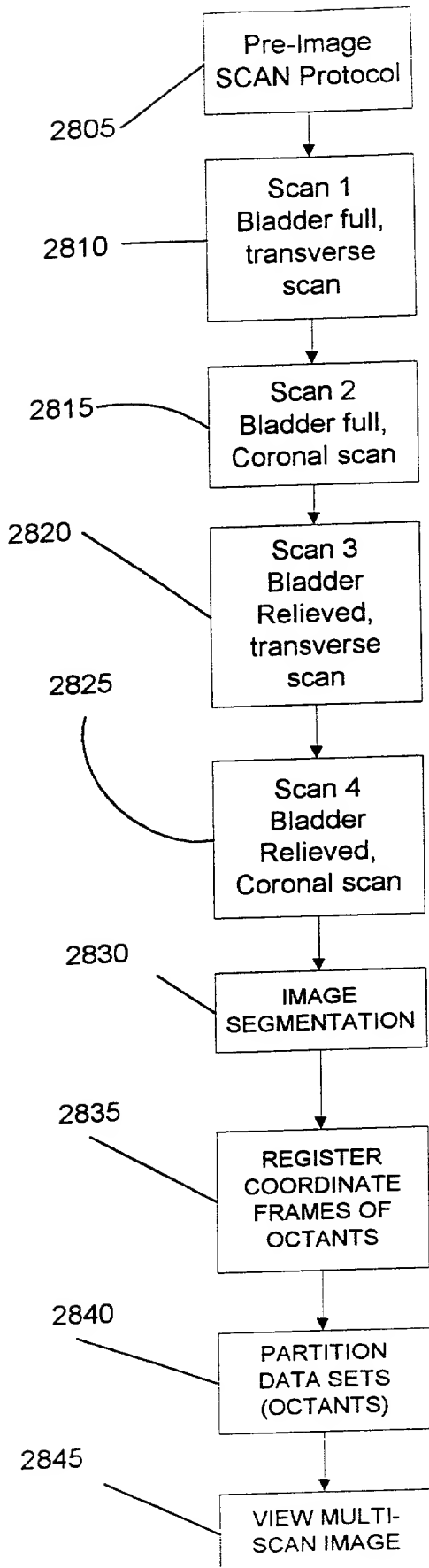


FIG. 26

Fig. 28



0977120, 051504

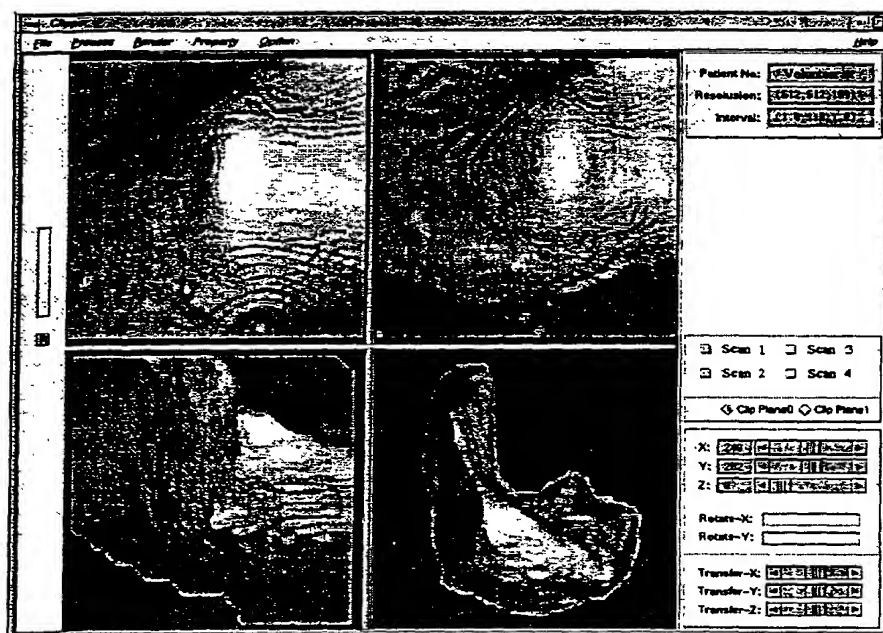
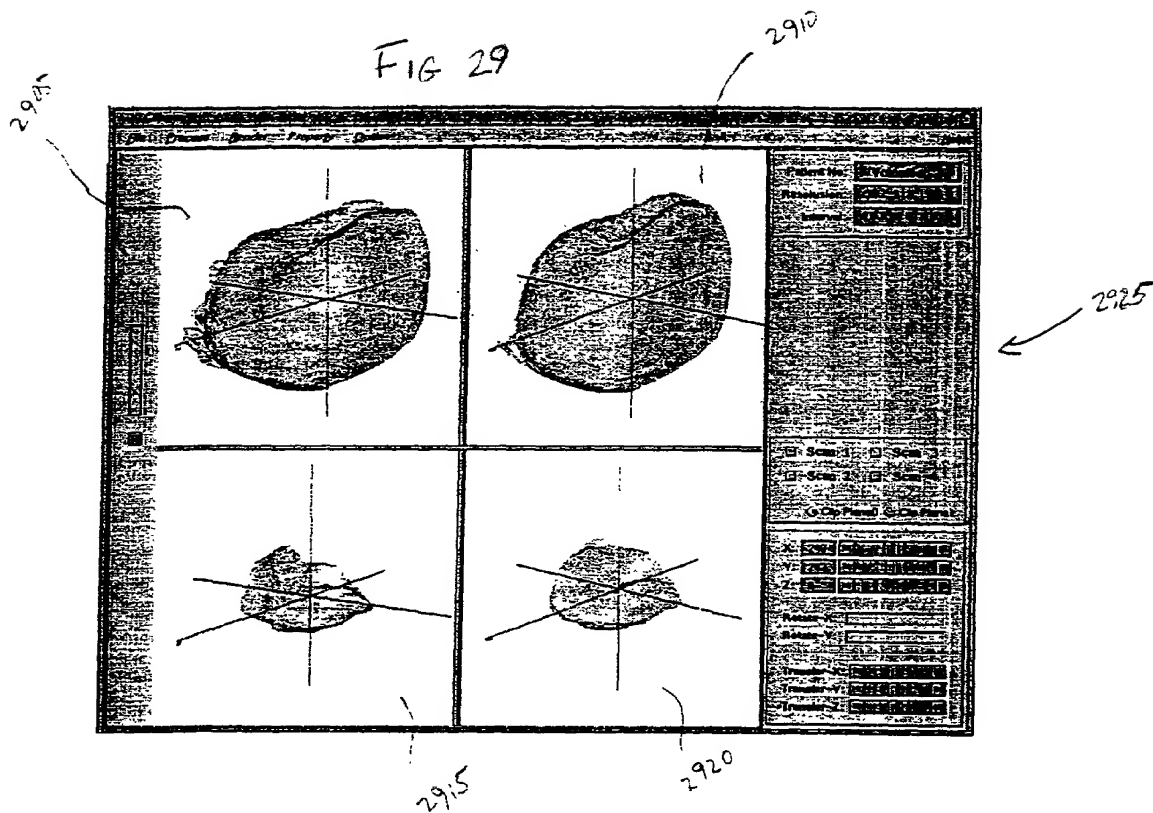


FIG. 31

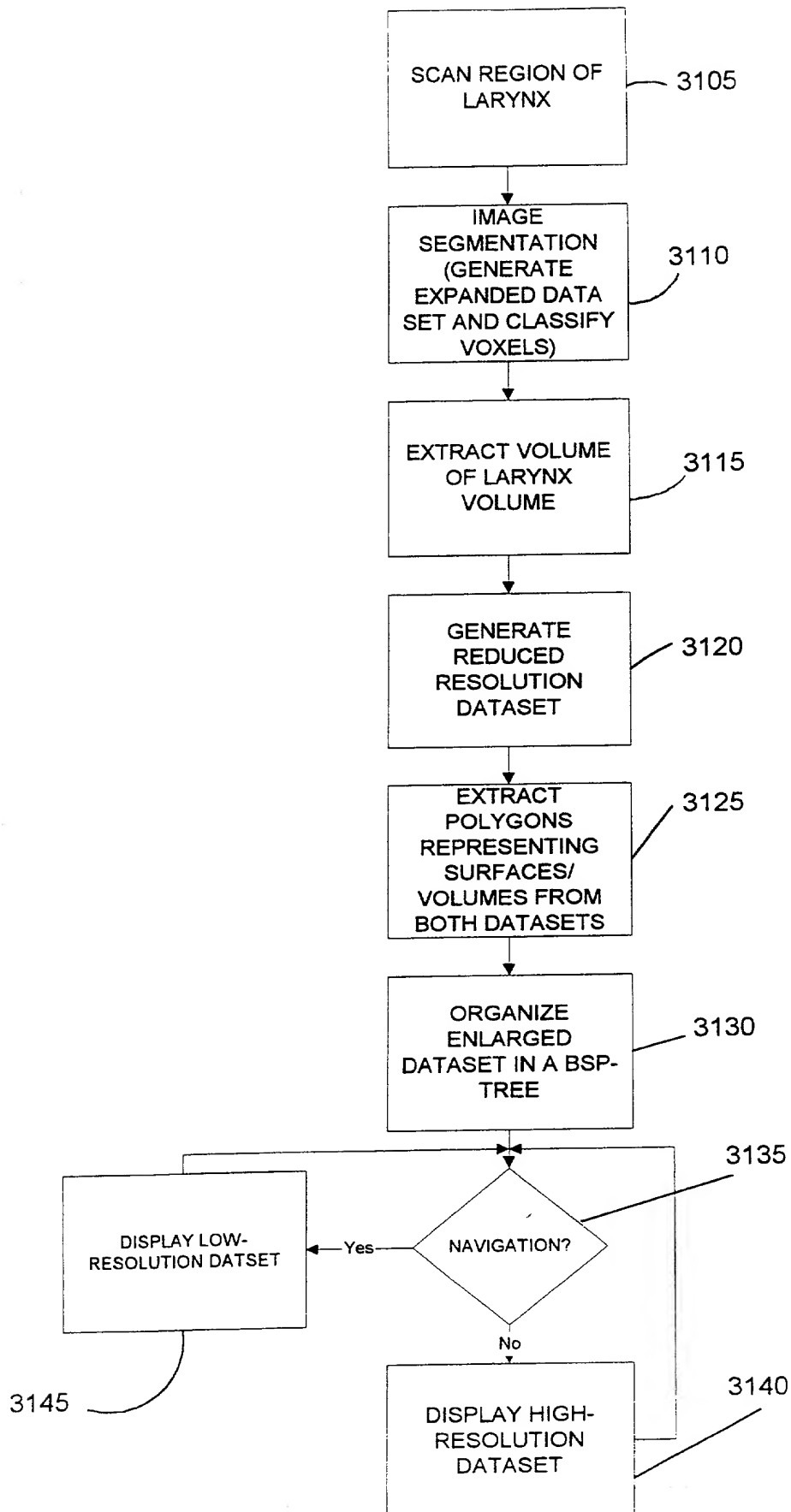


FIG. 32

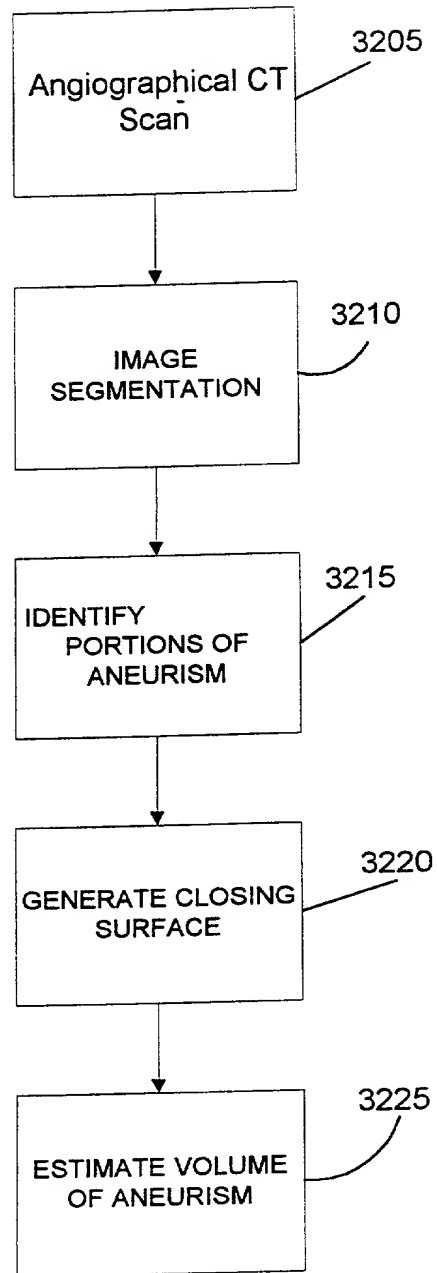


FIG. 33A

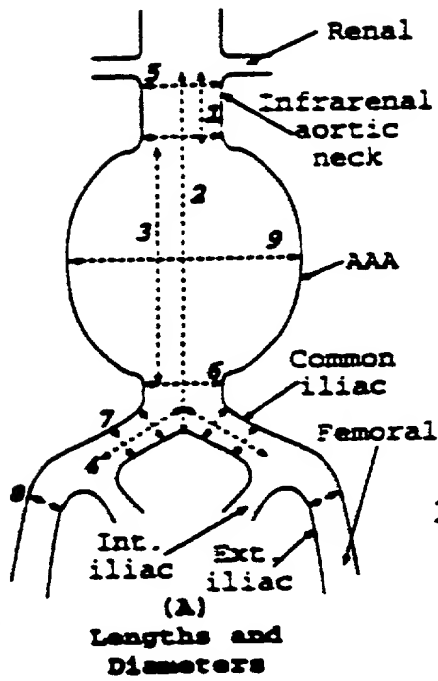


FIG. 33E

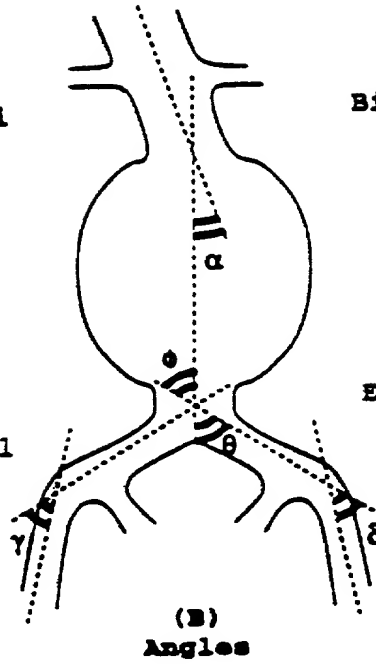


FIG 33C

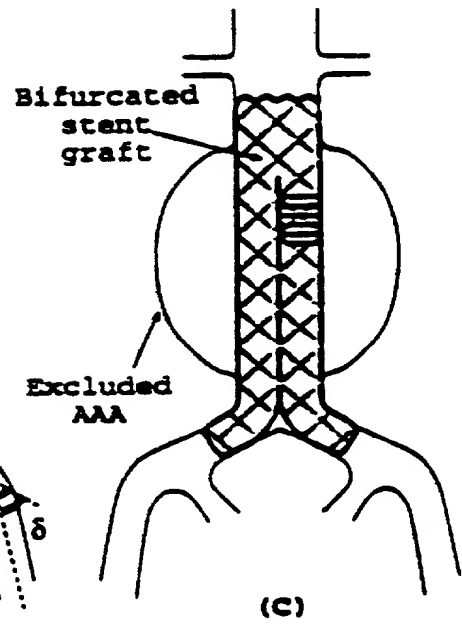


FIG. 34

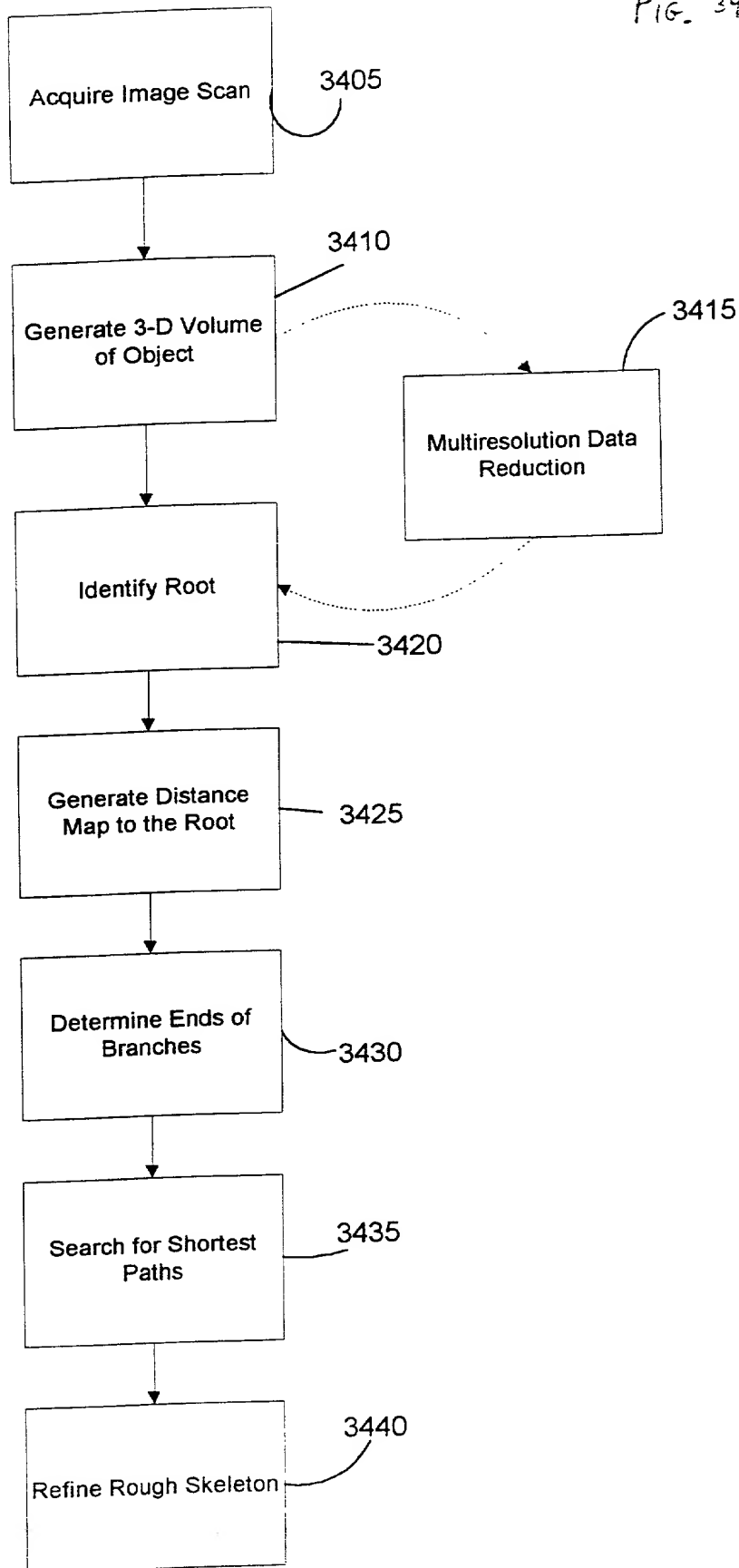


FIG. 35

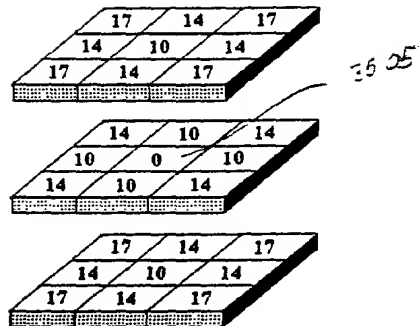


FIG. 36

```

1) Label root voxel with integer 0;
2) Construct a queue and line up the root in the queue;
3) If( There is at least one voxel in the queue )
    Serve the voxel x on the top of the queue:
    For( each of x's 26-connected neighbor voxel y ) {
        If( y in the volume and has not been labeled yet ) {
            Line up the y in the queue;

            /* label the voxel y */
            Set dist = 999999 ;
            For( each of y's 26-connected neighbor voxel z ) {
                If( z in the volume and has been already labeled with an integer of  $n_z$  ) {
                     $d_z \equiv n_z + d(y, z)$ ;
                    where  $d(y, z)$  is 10, 14 or 17 if the Euclidean distance between y and z is
                    1,  $\sqrt{2}$ , or  $\sqrt{3}$ , respectively;
                }
                If( dist >  $d_z$  ) {
                    label y with integer dist ;
                    dist =  $d_z$  ;
                }
            }
        }
    }
    x leaves the queue;
}
Else {
    end of calculating the distance map.
}

```